



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/026,681	12/18/2001	Muljadi Sulistio	CMRC 1009-1	7035

22470 7590 09/20/2006

HAYNES BEFFEL & WOLFELD LLP
P O BOX 366
HALF MOON BAY, CA 94019

EXAMINER

STEVENS, ROBERT

ART UNIT	PAPER NUMBER
----------	--------------

2162

DATE MAILED: 09/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.		Applicant(s)	
	10/026,681		SULISTIO ET AL.	
	Examiner		Art Unit	
	Robert Stevens		2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 June 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. The Office substantially maintains the previous rejections of the claims under 35 USC §§112-1st and 2nd paragraphs and §103(a), in light of the amendment. The Office, however, withdraws certain previous rejections of the claims under 35 USC §112-2nd paragraph, as indicated in the "Response to Arguments" section below.

Response to Arguments

2. Applicant's arguments have been fully considered but they are not persuasive.

Objection to Claims 16-16 and 23-24

The Office has withdrawn the objections to these claims, based upon the correction of minor informalities.

Rejection under 35 USC §101 of claims 1-22

Regarding claims 1 and 8, Applicant asserts on pages 8-9 of the Amendment, that the returning of results is not necessary for a software search component to be useful.

The Office respectfully disagrees. It stands to reason that if a search component merely searched, and provided no indication that it performed the intended search task, then it was of no use to the user/invoker of the search component. In fact, such a user has no way of knowing whether the component worked as intended, or even worked at all.

Regarding claim 15, Applicant asserts on page 9 of the Amendment, that the claim provides a useful, concrete and tangible result because the claimed GUI transforms data.

The Office respectfully disagrees. The asserted path specifications are merely "variables" that are assigned the value of the input data. This data is not transformed. It is merely *referenced* by the path specifications associated with the data entered into the claimed GUI. The filters merely control what GUI input fields are available for input, and thus provide no transformation functionality. Thus, the entered data is merely transferred from a data entry point to a server with no transformation of the data itself.

Regarding claim 16, Applicant asserts on page 9 of the Amendment, that if a filter may or may not exist, then the resulting claim is not directed to merely non-functional descriptive material.

The Office respectfully disagrees. There is no claimed functionality. The claims are directed to an arrangement of fields.

Regarding claims 2-7, 9-14 and 17-22, Applicant asserts on pages 9-10 of the Amendment, that the Examiner did not take into account each dependent claim in the rejection.

The Office respectfully disagrees. The Examiner did take each claim into account, and determined that none of the dependent claims add any limitation that

Art Unit: 2162

would render them statutory under 35 USC §101 using a useful, concrete and tangible analysis.

Rejection under 35 USC §112-1st paragraph of claims 5-7, 12-14 and 20-22

Regarding claims 5-7, 12-14 and 20-22, Applicant asserts on page 10 of the Amendment, that the previous rejections of the claims under 35 USC §112-1st paragraph are incorrect because a graphical user interface is implemented using an HTML or XML document, and that such a mark up language document is merely a character string.

The Office respectfully disagrees. Although comprised of text components, there is no way that a graphical user interface is implemented as merely a string. An HTML tag, for instance, may be represented as a string, but not a working/functional HTML document. As evidence, see the Applicant-supplied HTML file purportedly representing a version of the Deja Power Search GUI (pages 18-21 of the Amendment), and count the character string components of this file. A computer program may be coded as a collection of characters, but it's not merely a character string.

Rejection under 35 USC §112-2nd paragraph of claims 1-24

Regarding claims 1, 8, 15-16 and 23-24, Applicant asserts on pages 10-11 of the Amendment, that replacement of the word "the" with "respective" corrects a lack of antecedent basis for the terminology "document fields".

The Office respectfully disagrees. There still is no antecedent basis for the terminology "document fields". The Office notes that "document field selection filters" and "document fields" are not the same element.

Regarding claims 2-7, 9-14 and 17-22, Applicant asserts on pages 11 of the Amendment, that these claims, which depend upon claims 1, 8 and 16, are now acceptable under 35 USC §112-2nd paragraph.

The Office respectfully disagrees. See the rationale immediately above concerning claims 1, 8, 15-16 and 23-24.

Regarding claims 17-22, Applicant asserts on page 11 of the Amendment, that these claims were amended to recited dependence upon a graphical user interface parent claim.

The Office has withdrawn the rejections of these claims for this issue.

Regarding claims 5-7, 12-14 and 20-22, Applicant asserts on page 11 of the Amendment, that the rejection of these claims is improper because an XML or HTML document is a character string.

The Office respectfully disagrees. Although comprised of text components, there is no way that a graphical user interface is implemented via merely a string. An HTML tag, for instance, may be represented as a string, but not a working/functional HTML document. As evidence, see the Applicant-supplied HTML file representing a version of the Deja Power Search GUI (pages 18-21 of the Amendment), and count the character string components of this file. A computer program may be coded as a collection of characters, but it's not merely a character string.

Regarding claims 4, 11 and 19, Applicant asserts on page 11 of the Amendment, these claims were amended to recite dependence upon a different claim (rather than a claim that recited the same limitation).

The Office has withdrawn the rejections of these claims for this issue.

Rejection under 35 USC §103(a) of claims 1-4, 8-11, 15-19 and 23-24

Regarding claim 1, Applicant asserts on pages 12-15 of the Amendment, that the references do not teach context sensitive fields. On page 13, Applicant secondly asserts that the Altinel reference does not teach non-displaying fields. On page 13, Applicant thirdly further asserts that Altinel does not a GUI or a search engine. On page

13, Applicant fourthly asserts that since a dependent claim recites the use of XPATH for path specifications, it is improper to cite Altinel (for its XPATH teachings) vice the limitations independent claim 1. On page 14, Applicant fifthly makes a vague request for an affidavit covering the teachings of Altinel, which would indicate that these teachings are well known. On page 14, Applicant sixthly asserts that the motivation to combine the references is troubling. On page 14, Applicant seventhly references an appeal brief.

The Office respectfully disagrees. If search results are dependent upon field entries, those fields are, of necessity, context sensitive (i.e., if a field value were changed it would affect search results, which are affected by the *relationship between* or *context of* these fields/filters). Second, the concept of non-displaying or hidden GUI fields was well known in the art, and the use of such fields was an obvious variant considering the fact that a skilled artisan has the choice of two options, which perforce are obvious in light of each other: to displaying a field or not. Third, the Office notes that the references as a whole teach the well known concept of search engines or GUIs. Fourth, the Office asserts that art reading on the narrower dependent claim also reads on the broader independent claim. Fifth, Applicants desire for an affidavit regarding the teachings of Altinel is confusing, as the Altinel reference pre-dates Applicant's subject matter, and thus was well known in the art at the time of Applicant's subject matter. Sixth, the Office notes that, at the very least, the cited references are of the same field of endeavor, and thus properly combined. Seventh, the Office is unsure why Applicant has chosen to reference an irrelevant appeal brief, especially considering that the art

Art Unit: 2162

cited in the action being addressed by Applicant's amendment has been used in rejecting the claims for the first time.

Regarding claims 2-4, Applicant asserts on page 15 of the Amendment, that reciting the use of XML/XPATH renders the claims allowable.

The Office respectfully disagrees. XML and XPATH are well known programming languages. The mere use of a particular programming language does not impart patentability, as the choice of programming language to implement any functionality was merely an obvious variant to one skilled in the art at the time of the invention.

Regarding claim 8, Applicant asserts on page 15 of the Amendment, that a recitation which changes the processing location renders the claims allowable.

The Office respectfully disagrees. Applicant admits that this claim recites performance of the same functionality on a different processor. Moving processing among elements, software or hardware, was merely an obvious variant to one skilled in the art at the time of the invention. The same functionality exists, regardless of where that functionality is hosted.

Art Unit: 2162

Regarding claims 9-11, Applicant asserts on page 15 of the Amendment, that these claims are allowable for the reasons set forth concerning claims 2-4.

The Office respectfully disagrees. See the reasons set forth concerning claims 2-4.

Regarding claim 15, Applicant asserts on pages 15-16 of the Amendment, that this claim is allowable for the reasons set forth concerning claim 1.

The Office respectfully disagrees. See the reasons set forth concerning claim 1.

Regarding claim 16, Applicant appears to assert on page 16 of the Amendment, that this claim is allowable for the reasons set forth concerning claim 1. The actual argument states that claim **15** is similar to claim 1, and therefore claim **16** should be allowable.

The Office respectfully disagrees. See the reasons set forth concerning claim 1.

Regarding claims 17-19, Applicant asserts on page 16 of the Amendment, that these claims are allowable for the reasons set forth concerning claims 2-4.

The Office respectfully disagrees. See the reasons set forth concerning claims 2-4.

Art Unit: 2162

Regarding claims 23-24, Applicant asserts on page 16 of the Amendment, that a database is created before using a GUI to search that database, and therefore the claims are allowable.

The Office respectfully disagrees. The searchable entity, i.e., database must exist before querying that entity for search results. It is unclear what Applicant believes to be inventive.

Rejection under 35 USC §103(a) of claims 5-7, 12-14 and 20-22

Regarding claim 5, Applicant asserts on pages 16-17 of the Amendment, that the claims are allowable because an HTML document is a character string, it is well known to implement GUIs using HTML, and for the reasons asserted regarding claim 1.

The Office respectfully disagrees. As set forth both above and below, an HTML document (at least a valid, working one) is not merely a string. Although it is well known to implement GUIs in HTML, that doesn't impart allowability to the asserted claims. See the Office's assertions regarding claim 1.

Regarding claims 6-7, 12-14 and 20-22, Applicant asserts on page 17 of the Amendment, these claims are allowable for the reasons set forth concerning claim 5.

The Office respectfully disagrees. See the reasons set forth concerning claim 5.

For these reasons, the Office asserts the rejections of the claims as set forth below.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. **Claims 1-22 are rejected under 35 U.S.C. 101** because the claimed invention is directed to non-statutory subject matter.

Regarding independent claims 1 and 8: These claims “provide” a graphical user interface (GUI), receive a couple of values, and then search documents based on those values. No useful, concrete and tangible result is accomplished. For example, there is no providing of results to constitute a tangible result, so as to be able to realize the usefulness of a search.

Regarding independent claim 15: This claim “provides” a GUI, as in claim 1, receives user input, and forwards that user input to a server. Although arguably tangible, the transmission of data provides no concrete or useful result. This data is merely moved from point A to point B. There is no transformation, for instance, that takes place. This claim thus produces no useful, concrete and tangible result.

Regarding independent claim 16: This claim is directed to a “computer-implemented” GUI. Although tangibly embodied, a graphical user interface is merely an

arrangement of fields, and therefore non-statutory under 35 USC 101 as being non-functional descriptive material. The Office interprets the recited "filters" as merely fields for data entry. The data is intended for use in a filtering process, presumably implemented in an unclaimed search engine functionality.

Claims 2-7, 9-14 and 17-22 depend upon claims 1, 8 and 16, respectively, and are therefore likewise rejected.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. **Claims 5-7, 12-14 and 20-22 are rejected under 35 U.S.C. 112, first paragraph**, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Regarding claims 5-7, 12-14 and 20-22, there is no enabling disclosure in the application as to how one would implement the graphical user interface, as claimed in the parent independent claim of each these dependent claims, as a "character string".

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. **Claims 1-24 are rejected under 35 U.S.C. 112, second paragraph**, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Independent claims 1 (lines 8-9), 8 (lines 8-9), 15 (lines 8-9), 16 (lines 6-7), 23 (lines 11-12) and 24 (lines 11-12) each recite the limitation "respective document fields" in lines 8-9. There is insufficient antecedent basis for this limitation in the claim. It is noted that, in each case, the preceding GUI limitation is actually directed to a selection filter, and not a document field.

Claims 2-7, 9-14 and 17-22 depend upon claims 1, 8 and 16, respectively, and are therefore likewise rejected.

Further regarding claims 5-7, 12-14 and 20-22, it is unclear what is being claimed. The parent independent claim associated with each of these dependent claims recites a document type selection filter, field selection filter, specification fields and hidden fields. It is unclear how a mere character string, as recited in claims 5-7, 12-14 and 20-22 could represent any of those limitations (especially that of a hidden or non-displaying field) recited in each of the corresponding parent independent claims.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. **Claims 1-4, 8-11, 15-19 and 23-24 are rejected under 35 U.S.C. 103(a)** as being unpatentable over the Deja Power Search Graphical User Interface Form ("Deja Power Search Graphical User Interface", downloaded from: www.exit109.com/~jeremy/news/deja.html, © Feb. 12, 2000, pp. 1-20, hereafter referred to as "DPSGUI") in view of Mehmet Altinel et al. ("Efficient Filtering of XML Documents for Selective Dissemination of Information", Proceedings of the VLDB Conference, Cairo, Egypt, Sep. 10-14, 2000, pp. 53-64, hereafter referred to as "Altinel").

Regarding independent claim 1, DPSGUI discloses: ***A computer-implemented method of searching a plurality of self-describing, structured documents, said documents including document fields, the method including: providing a graphical user interface including: a document type selection filter; one or more document field selection filters, context sensitive to a selected document type; one or more value specification fields, context sensitive to respective document fields;*** (See DPSGUI pages 1-4, showing a graphical user interface comprised of several fields for inputting search terms and selecting field filters. This GUI enabled a user to input data used for defining searches. The DPSGUI comprises a drop down selector field labeled "Archive" for selecting a document type, such as "complete" (all document types, see page 1 "Archive" field), "jobs" (employment document types, see page 2 "Archive" field), "for sale" (classified/want ad document types, see page 3 "Archive" field), and "adult" (adult content document types, see page 4 "Archive" field). DPSGUI further discloses a filtering capability based upon a "Subject" and/or "Newsgroup" using the like-named TextFields shown in the GUI of page 1. The DPSGUI further indicates that these document field selection filters may be saved. DPSGUI additionally comprises a TextField allowing a user to input "Keywords" (See the TextField labeled "Keywords" on page 1), which have a certain value. It is implicit that the inputs to the DPSGUI are all utilized in a requested document search, when a user invokes one of the "Search" buttons found on the GUI of page 1, for example. In

Art Unit: 2162

other words, all selected inputs are utilized in the search query, and thus are utilized in the context of each other to limit/filter the results of a search query.)

Although DPSGUI provides a user interface for accepting document types and value specifications and kicks off a search of documents upon user selection of a "Search" button (see page 1 "Search" button), DPSGUI does not explicitly disclose using XPath expressions in the search of self-describing, structured documents, such as XML documents. Altinel, though, discloses: ***as non-displaying fields, one or more path specifications corresponding to the document fields and to the value specification fields, said path specifications identifying nodes to be tested against completed value specifications; receiving the selected document type and the completed value specifications and the corresponding path specifications; and searching a subset of the self-describing, structured documents based on the completed value specifications and the corresponding path specifications, the subset including documents of the selected document type.*** (See Altinel disclosing the well-known use of XML documents in the last paragraph of page 53 ["We have developed...."]. Altinel further discloses the well-known use of XPath for searching in the first and second paragraphs on page 55 [both paragraphs are found above the section entitled "3 Related Work"]. Altinel discusses the use of the XPath expression "//product[price/msrp<300]/name" in the first paragraph to locate name elements in a XML document if the msrp of the product is less than 300. In the second paragraph, Altinel further discloses the selection of a XML document if an

Art Unit: 2162

XPath expression matched at least one element of that document. It was implied that if Altinel used an XPath expression, that Altinel received that expression. It is noted that the choices as to whether or not to perform an action [i.e., to display a field or not] were obvious in light of each other, because, after having chosen a display option, one skilled in the art would have been aware that the other option existed and was therefore an available choice.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Altinel for the benefit of DPSGUI, because to do so would have enabled a programmer to develop a document filtering system that provided highly efficient matching of XML documents to large numbers of user profiles, as taught by Altinel in the last paragraph of page 53 ("We have developed...."). These references were all applicable to the same field of endeavor, i.e., electronic document searching.

Regarding claims 2-4, DPSGUI does not explicitly mandate the use of any particular mark up language in coding the GUI of page 1. Altinel, though, discloses the well known use of XML and XPATH. (See Altinel in the last paragraph of page 53 ["We have developed...."]. Altinel further discloses the well-known use of XPath in the first and second paragraphs on page 55 [both paragraphs are found above the section entitled "3 Related Work"]. Altinel discusses the use of the XPath expression "//product[price/msrp<300]/name" in the first paragraph to locate name elements in a XML document if the msrp of the product is less than 300. In the second paragraph,

Altinel further discloses the selection of a XML document if an XPath expression matches at least one element of that document.)

Regarding independent claim 8, DPSGUI discloses: **A computer-implemented method of searching a plurality of self-describing, structured documents, said documents including document fields, the method including: providing a graphical user interface including: *a document type selection filter; one or more document field selection filters, context sensitive to a selected document type; and one or more value specification fields, context sensitive to respective document fields;*** (See DPSGUI pages 1-4, showing a graphical user interface comprised of several fields for inputting search terms and selecting filed filters. This GUI enabled a user to input data used for defining searches. The DPSGUI comprises a drop down selector field labeled "Archive" for selecting a document type, such as "complete" (all document types, see page 1 "Archive" field), "jobs" (employment document types, see page 2 "Archive" field), "for sale" (classified/want ad document types, see page 3 "Archive" field), and "adult" (adult content document types, see page 4 "Archive" field). DPSGUI further discloses a filtering capability based upon a "Subject" and/or "Newsgroup" using the like-named TextFields shown in the GUI of page 1. The DPSGUI further indicates that these document field selection filters may be saved. DPSGUI additionally comprises a TextField allowing a user to input "Keywords" (See the TextField labeled "Keywords" on page 1), which have a certain value. It is implicit that the inputs to the DPSGUI are all utilized in a requested document search, when a

Art Unit: 2162

user invokes one of the "Search" buttons found on the GUI of page 1, for example. In other words, all selected inputs are utilized in the search query, and thus are utilized in the context of each other to limit/filter the results of a search query.)

Although DPSGUI provides a user interface for accepting document types and value specifications and kicks off a search of documents upon user selection of a "Search" button (see page 1 "Search" button), DPSGUI does not explicitly disclose using XPath expressions in the search of self-describing, structured documents, such as XML documents. Altinel, though, discloses: ***receiving the selected document type and the completed value specifications and document field identifiers corresponding to the completed value specifications; looking up path specifications corresponding to the document field identifiers, said paths specifications identifying nodes to be tested against completed value specifications; and searching a subset of the self-describing, structured documents based on the completed value specifications and the corresponding path specifications, the subset including documents of the selected document type.*** (See Altinel disclosing the well-known use of XML documents in the last paragraph of page 53 ["We have developed...."]. Altinel further discloses the well-known use of XPath for searching in the first and second paragraphs on page 55 [both paragraphs are found above the section entitled "3 Related Work"]. Altinel discusses the use of the XPath expression "//product[price/msrp<300]/name" in the first paragraph to locate name elements in a XML document if the msrp of the product is less than 300.

Art Unit: 2162

In the second paragraph, Altinel further discloses the selection of a XML document if an XPath expression matched at least one element of that document. It was implied that if Altinel used an XPath expression, that Altinel received that expression. It is noted that the choices as to whether or not to perform an action [i.e., to display a field or not] were obvious in light of each other, because, after having chosen a display option, one skilled in the art would have been aware that the other option existed and was therefore an available choice. Altinel further teaches that a XML document is abstracted as a tree of nodes, and that XPath expressions are patterns that can be matched to nodes in the XML tree. See the first paragraph under the section entitled "2.2 XPath as a Profile Language" on page 54.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Altinel for the benefit of DPSGUI, because to do so would have enabled a programmer to develop a document filtering system that provided highly efficient matching of XML documents to large numbers of user profiles, as taught by Altinel in the last paragraph of page 53 ("We have developed...."). These references were all applicable to the same field of endeavor, i.e., electronic document searching.

Claims 9-11 are substantially similar to claims 2-4, and are therefore likewise rejected.

Regarding independent claim 15, DPSGUI discloses: ***A method of specifying where to search among a plurality of self-describing, structured documents, said documents having document types and including document fields, the method including: displaying a graphical user interface including: a document type selection filter; one or more document field selection filters, context sensitive to a selected document type; and one or more value specification fields, context sensitive to respective document fields;*** (See DPSGUI pages 1-4, showing a graphical user interface comprised of several fields for inputting search terms and selecting field filters. This GUI enabled a user to input data used for defining searches. The DPSGUI comprises a drop down selector field labeled "Archive" for selecting a document type, such as "complete" (all document types, see page 1 "Archive" field), "jobs" (employment document types, see page 2 "Archive" field), "for sale" (classified/want ad document types, see page 3 "Archive" field), and "adult" (adult content document types, see page 4 "Archive" field). DPSGUI further discloses a filtering capability based upon a "Subject" and/or "Newsgroup" using the like-named TextFields shown in the GUI of page 1. The DPSGUI further indicates that these document field selection filters may be saved. DPSGUI additionally comprises a TextField allowing a user to input "Keywords" (See the TextField labeled "Keywords" on page 1), which have a certain value. It is implicit that the inputs to the DPSGUI are all utilized in a requested document search, when a user invokes one of the "Search" buttons found on the GUI of page 1, for example. In other words, all selected inputs are

Art Unit: 2162

utilized in the search query, and thus are utilized in the context of each other to limit/filter the results of a search query.)

Although the DPSGUI does teach a graphical user interface that does not display XPath specifications, provides a user interface for accepting document types and value specifications, kicks off a search of documents upon user selection of a "Search" button (see page 1 "Search" button), and the implied transmission of GUI input to a server (see text at bottom of GUI labeled as "Fine print", which teaches the use of an Internet Service Provider [ISP]), DPSGUI does not explicitly disclose using XPath expressions in the search of self-describing, structured documents, such as XML documents. Altinel, though, discloses: ***the graphical user interface further including, as non-displaying fields, one or more path specifications corresponding to the document fields and to the value specification fields, said path specifications identifying nodes in the documents to be tested against completed value specifications; receiving from a user the selected document type and the completed value specifications; and transmitting to a server the selected document type and the completed value specifications and the path specifications corresponding to the completed value specifications.*** (See Altinel disclosing the well-known use of XML documents in the last paragraph of page 53 ["We have developed...."]. Altinel further discloses the well-known use of XPath for searching in the first and second paragraphs on page 55 [both paragraphs are found above the section entitled "3 Related Work"]. Altinel discusses the use of the XPath expression "//product[price/msrp<300]/name" in the first paragraph

Art Unit: 2162

to locate name elements in a XML document if the msrp of the product is less than 300. In the second paragraph, Altinel further discloses the selection of a XML document if an XPath expression matched at least one element of that document. It was implied that if Altinel used an XPath expression, that Altinel received that expression. It is noted that the choices as to whether or not to perform an action [i.e., to display a field or not] were obvious in light of each other, because, after having chosen a display option, one skilled in the art would have been aware that the other option existed and was therefore an available choice.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Altinel for the benefit of DPSGUI, because to do so would have enabled a programmer to develop a document filtering system that provided highly efficient matching of XML documents to large numbers of user profiles, as taught by Altinel in the last paragraph of page 53 ("We have developed...."). These references were all applicable to the same field of endeavor, i.e., electronic document searching.

Regarding independent claim 16, DPSGUI discloses: ***A computer-implemented graphical user interface, including: a document type selection filter; one or more document field selection filters, context sensitive to a selected document type; one or more value specification fields, context sensitive to respective document fields;*** (See DPSGUI pages 1-4, showing a graphical user interface comprised of several fields for inputting search terms and selecting field filters. This GUI enabled a user to input data used for defining searches. The DPSGUI

Art Unit: 2162

comprises a drop down selector field labeled "Archive" for selecting a document type, such as "complete" (all document types, see page 1 "Archive" field), "jobs" (employment document types, see page 2 "Archive" field), "for sale" (classified/want ad document types, see page 3 "Archive" field), and "adult" (adult content document types, see page 4 "Archive" field). DPSGUI further discloses a filtering capability based upon a "Subject" and/or "Newsgroup" using the like-named TextFields shown in the GUI of page 1. The DPSGUI further indicates that these document field selection filters may be saved. DPSGUI additionally comprises a TextField allowing a user to input "Keywords" (See the TextField labeled "Keywords" on page 1), which have a certain value. It is implicit that the inputs to the DPSGUI are all utilized in a requested document search, when a user invokes one of the "Search" buttons found on the GUI of page 1, for example. In other words, all selected inputs are utilized in the search query, and thus are utilized in the context of each other to limit/filter the results of a search query.)

Although DPSGUI provides a user interface for accepting document types and value specifications and kicks off a search of documents upon user selection of a "Search" button (see page 1 "Search" button), DPSGUI does not explicitly disclose using XPath expressions in the search of self-describing, structured documents, such as XML documents. Altinel, though, discloses: ***as non-displaying fields, one or more path specifications corresponding to the document fields and to the value specification fields, said path specifications identifying nodes of a self-describing, structured document to be tested against completed value***

specifications. (See Altinel disclosing the well-known use of XML documents in the last paragraph of page 53 ["We have developed...."]. Altinel further discloses the well-known use of XPath for searching in the first and second paragraphs on page 55 [both paragraphs are found above the section entitled "3 Related Work"]. Altinel discusses the use of the XPath expression "//product[price/msrp<300]/name" in the first paragraph to locate name elements in a XML document if the msrp of the product is less than 300. In the second paragraph, Altinel further discloses the selection of a XML document if an XPath expression matched at least one element of that document. It was implied that if Altinel used an XPath expression, that Altinel received that expression. It is noted that the choices as to whether or not to perform an action [i.e., to display a field or not] were obvious in light of each other, because, after having chosen a display option, one skilled in the art would have been aware that the other option existed and was therefore an available choice.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Altinel for the benefit of DPSGUI, because to do so would have enabled a programmer to develop a document filtering system that provided highly efficient matching of XML documents to large numbers of user profiles, as taught by Altinel in the last paragraph of page 53 ("We have developed...."). These references were all applicable to the same field of endeavor, i.e., electronic document searching.

Claims 17-19 are substantially similar to claims 2-4, and are therefore likewise rejected.

Regarding independent claim 23, DPSGUI discloses: ***A method of providing a searchable data base of self-describing, structured documents, including: providing a graphical user interface based on the set, including: a document type selection filter; one or more document field selection filters, context sensitive to a selected document type; one or more value specification fields, context sensitive to respective document fields;*** (See DPSGUI pages 1-4, showing a graphical user interface comprised of several fields for inputting search terms and selecting filed filters. This GUI enabled a user to input data used for defining searches. The DPSGUI comprises a drop down selector field labeled "Archive" for selecting a document type, such as "complete" (all document types, see page 1 "Archive" field), "jobs" (employment document types, see page 2 "Archive" field), "for sale" (classified/want ad document types, see page 3 "Archive" field), and "adult" (adult content document types, see page 4 "Archive" field). DPSGUI further discloses a filtering capability based upon a "Subject" and/or "Newsgroup" using the like-named TextFields shown in the GUI of page 1. The DPSGUI further indicates that these document field selection filters may be saved. DPSGUI additionally comprises a TextField allowing a user to input "Keywords" (See the TextField labeled "Keywords" on page 1), which have a certain value. It is implicit that the inputs to the DPSGUI are all utilized in a requested document search, when a

Art Unit: 2162

user invokes one of the “Search” buttons found on the GUI of page 1, for example. In other words, all selected inputs are utilized in the search query, and thus are utilized in the context of each other to limit/filter the results of a search query.)

Although DPSGUI provides a user interface for accepting document types and value specifications and kicks off a search of documents upon user selection of a “Search” button (see page 1 “Search” button), DPSGUI does not explicitly disclose using XPath expressions in the search of self-describing, structured documents, such as XML documents, and the loading and indexing of documents. Altinel, though, discloses: ***as non-displaying fields, one or more path specifications corresponding to the document fields and to the value specification fields, said path specifications identifying nodes of a self-describing, structured document to be tested against completed value specifications.*** (See Altinel disclosing the well-known use of XML documents in the last paragraph of page 53 [“We have developed....]. Altinel further discloses the well-known use of XPath for searching in the first and second paragraphs on page 55 [both paragraphs are found above the section entitled “3 Related Work”]. Altinel discusses the use of the XPath expression “//product[price/msrp<300]/name” in the first paragraph to locate name elements in a XML document if the msrp of the product is less than 300. In the second paragraph, Altinel further discloses the selection of a XML document if an XPath expression matched at least one element of that document. It was implied that if Altinel used an XPath expression, that Altinel received that expression. It is noted that the choices as

Art Unit: 2162

to whether or not to perform an action [i.e., to display a field or not] were obvious in light of each other, because, after having chosen a display option, one skilled in the art would have been aware that the other option existed and was therefore an available choice. Also see Figure 3 of page 57, especially Figure 3a, and Figure 5 of page 59 further disclose indexing of portions of XML documents based upon XPath queries. It was further implied that document search parameters must have been loaded in order for XPath processing to have taken place. See the first paragraph under the section entitled "2.2 XPath as a Profile Language" on page 54 in context of the first paragraph on page 55, which discusses XPath processing and sets forth an XPath query `"//product[price/msrp<300]/name"` for selecting name elements of the XML document if the msrp of the product is less than 300.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Altinel for the benefit of DPSGUI, because to do so would have enabled a programmer to develop a document filtering system that provided highly efficient matching of XML documents to large numbers of user profiles, as taught by Altinel in the last paragraph of page 53 ("We have developed...."). These references were all applicable to the same field of endeavor, i.e., electronic document searching.

Regarding independent claim 24, DPSGUI discloses: A method of providing a searchable data base of self-describing, structured documents, including: providing a graphical user interface including a document type selection filter; one or more

Art Unit: 2162

document field selection filters, context sensitive to a selected document type; one or more value specification fields, context sensitive to the document fields; (See DPSGUI pages 1-4, showing a graphical user interface comprised of several fields for inputting search terms and selecting field filters. This GUI enabled a user to input data used for defining searches. The DPSGUI comprises a drop down selector field labeled "Archive" for selecting a document type, such as "complete" (all document types, see page 1 "Archive" field), "jobs" (employment document types, see page 2 "Archive" field), "for sale" (classified/want ad document types, see page 3 "Archive" field), and "adult" (adult content document types, see page 4 "Archive" field). DPSGUI further discloses a filtering capability based upon a "Subject" and/or "Newsgroup" using the like-named TextFields shown in the GUI of page 1. The DPSGUI further indicates that these document field selection filters may be saved. DPSGUI additionally comprises a TextField allowing a user to input "Keywords" (See the TextField labeled "Keywords" on page 1), which have a certain value. It is implicit that the inputs to the DPSGUI are all utilized in a requested document search, when a user invokes one of the "Search" buttons found on the GUI of page 1, for example. In other words, all selected inputs are utilized in the search query, and thus are utilized in the context of each other to limit/filter the results of a search query.)

Although DPSGUI provides a user interface for accepting document types and value specifications and kicks off a search of documents upon user selection of a "Search" button (see page 1 "Search" button), DPSGUI does not explicitly disclose

Art Unit: 2162

using XPath expressions in the search of self-describing, structured documents, such as XML documents, and the loading and indexing of documents. Altinel, though, discloses: ***as non-displaying fields, one or more path specifications corresponding to the document fields and to the value specification fields, said path specifications identifying nodes of a self-describing, structured document to be tested against completed value specifications.*** (See Altinel disclosing the well-known use of XML documents in the last paragraph of page 53 [“We have developed....]. Altinel further discloses the well-known use of XPath for searching in the first and second paragraphs on page 55 [both paragraphs are found above the section entitled “3 Related Work”]. Altinel discusses the use of the XPath expression “//product[price/msrp<300]/name” in the first paragraph to locate name elements in a XML document if the msrp of the product is less than 300. In the second paragraph, Altinel further discloses the selection of a XML document if an XPath expression matched at least one element of that document. It was implied that if Altinel used an XPath expression, that Altinel received that expression. It is noted that the choices as to whether or not to perform an action [i.e., to display a field or not] were obvious in light of each other, because, after having chosen a display option, one skilled in the art would have been aware that the other option existed and was therefore an available choice. Also see Figure 3 of page 57, especially Figure 3a, and Figure 5 of page 59 further disclose indexing of portions of XML documents based upon XPath queries. It was further implied that document search parameters must have been loaded in order for XPath processing to have taken place. See the first paragraph under the section

Art Unit: 2162

entitled "2.2 XPath as a Profile Language" on page 54 in context of the first paragraph on page 55, which discusses XPath processing and sets forth an XPath query `"//product[price/msrp<300]/name"` for selecting name elements of the XML document if the msrp of the product is less than 300.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Altinel for the benefit of DPSGUI, because to do so would have enabled a programmer to develop a document filtering system that provided highly efficient matching of XML documents to large numbers of user profiles, as taught by Altinel in the last paragraph of page 53 ("We have developed...."). These references were all applicable to the same field of endeavor, i.e., electronic document searching.

11. Claims 5-7, 12-14, and 20-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Deja Power Search Graphical User Interface Form ("Deja Power Search Graphical User Interface", downloaded from: www.exit109.com/~jeremy/news/deja.html, © Feb. 12, 2000, pp. 1-20, hereafter referred to as "DPSGUI") in view of Mehmet Altinel et al. ("Efficient Filtering of XML Documents for Selective Dissemination of Information", Proceedings of the VLDB Conference, Cairo, Egypt, Sep. 10-14, 2000, pp. 53-64, hereafter referred to as "Altinel") and further in view of Adar et al. (US Patent No. 6,493,702, filed May 5, 1995 and issued Dec. 10, 2002, hereafter referred to as "Adar").

Regarding claim 5, DPSGUI does not explicitly disclose that its graphical user interface was implemented in HTML. Adar, though, discloses the well-known use of HTML to implement graphical user interfaces. (See Adar Fig. 4 #410 and Fig. 6 #610, taken in context of Fig. 9 #914, showing an HTML interpreter for a user's browser. Adar further discloses the use of a browser to interpret HTML pages in col. 10 lines 12-21.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Adar for the benefit of DPSGUI in view of Altinel, because to do so would have enabled a programmer to develop a system that employed user and group profiles to augment Internet searches, re-rank search results, and provide recommendations for documents based on a subject-matter query, as taught by Adar in the Abstract. These references were all applicable to the same field of endeavor, i.e., electronic document searching.

Claims 6-7, 12-14 and 20-22 are each substantially similar to claim 5, and are therefore likewise rejected.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Non-Patent Literature

"Deja Power Search Graphical User interface", downloaded from: web.archive.org/web/19991008231252/http://www.exit109.com/~jeremy/news/deja.html, dated Oct. 8, 1999, pp. 1-20.

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

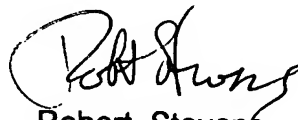
Art Unit: 2162

Contact Information

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert Stevens whose telephone number is (571) 272-4102. The examiner can normally be reached on M-F 6:00 - 2:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



Robert Stevens
Examiner
Art Unit 2162

September 13, 2006



SHAHID ALAM
PRIMARY EXAMINER